Recent Improvements in the ATLAS PanDA Pilot

P. Nilsson[1*], J. Caballero Bejar[2], G. Compostella[3], C. Contreras[4], K. De[1], T. Dos Santos[5], T. Maeno[2], M. Potekhin[2], T. Wenaus[2]
on behalf of the ATLAS Collaboration
[1]University of Texas at Arlington, TX, USA
[2]Brookhaven National Laboratory, NY, USA
[3]Max-Planck-Institut für Physik, München, Germany
[4]Departamento de Fisica, Univ. Tecnica Federico Santa Maria, Valparaiso, Chile
[5]Bergische Universität, Wuppertal, Germany
Corresponding author: paul.nilsson@cern.ch

Abstract. The Production and Distributed Analysis system (PanDA) in the ATLAS experiment uses pilots to execute submitted jobs on the worker nodes. The pilots are designed to deal with different runtime conditions and failure scenarios, and support many storage systems. This talk will give a brief overview of the PanDA pilot system and will present major features and recent improvements including CernVM File System integration, the job retry mechanism, advanced job monitoring including JEM technology, and validation of new pilot code using the HammerCloud stress-testing system. PanDA is used for all ATLAS distributed production and is the primary system for distributed analysis. It is currently used at over 100 sites worldwide. We analyze the performance of the pilot system in processing LHC data on the OSG, EGI and Nordugrid infrastructures used by ATLAS, and describe plans for its further evolution.

1. Introduction

A generic approach in grid computing is to use pilot jobs. Instead of submitting jobs directly to the grid gatekeepers, pilot factories are used to submit special lightweight jobs referred to here as pilot wrappers that are executed on the worker nodes. In the case of ATLAS [1], the pilot wrappers perform initial checks, download the main pilot code and launch it [2, 3]. The pilot is responsible for downloading the actual payload and any input files from the Storage Element (SE), executing the payload, uploading the output to the SE, and sending the final job status to the PanDA server.

PanDA [4], *Production and Distributed Analysis*, is the pilot-based workload management system used by the ATLAS Experiment. It was developed to meet the needs of ATLAS distributed computing and has been in production since 2005. The PanDA system has proven to be very successful in managing the ATLAS distributed production and analysis requirements on all three ATLAS grids; OSG [5], EGI [6] and Nordugrid [7]. Today PanDA is being used to incorporate cloud computing resources into ATLAS computing [8].

2. Pilot workflow

Upon launch, the pilot starts with performing initial setups and creates a signal handler that can take care of incoming signals from the batch or operating system [Fig. 1]. The pilot can optionally recover found remnants of previously failed jobs. It then proceeds with entering the main pilot loops. Before the pilot downloads a job, special criteria have to be met; enough local resources have to be available and the lifetime of the proxy must be long enough to last until the end of the current job. For each job a sub process is forked that is monitored by the pilot until it ends. The sub process determines the proper setup for the job before any input is transferred. It executes the payload and transfers the output at the end of the job. When the sub process has finished, the pilot archives the work directory and transfers it to the SE, before sending the final job status to the PanDA server.

---

* To whom any correspondence should be addressed.

The progress of a job can be followed on the PanDA monitor [9], which also has a link to the full job log archive at the end of the job. In multi-job mode, the pilot runs sequential jobs until it runs out of a predetermined time.

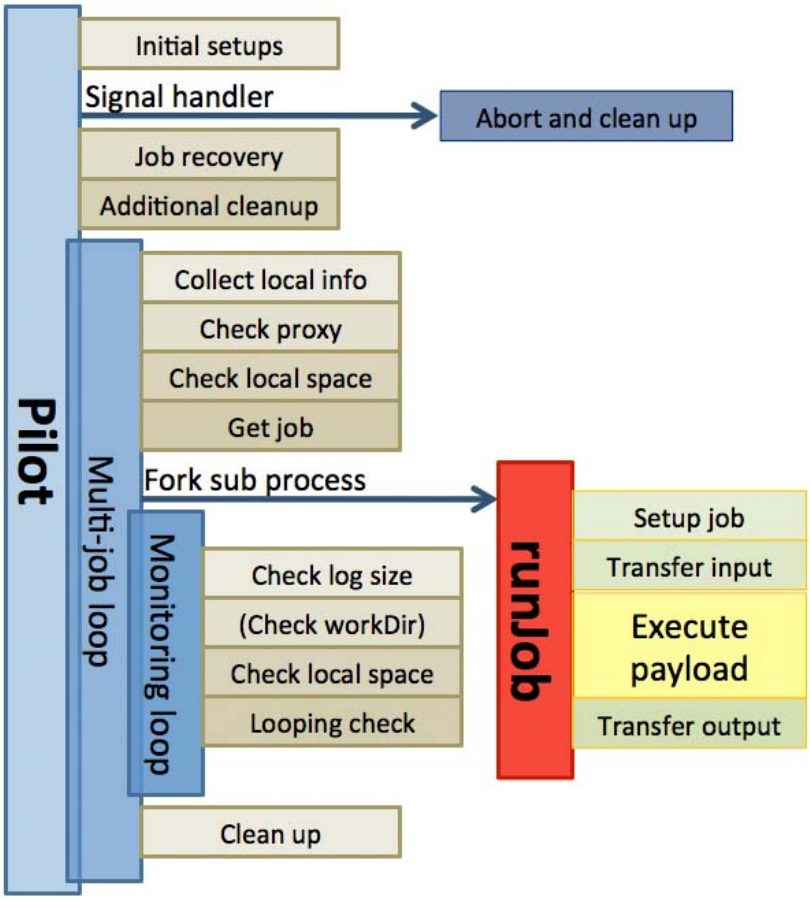

Figure 1. Pilot workflow.

3. Next generation pilot

In the next generation pilot, which is currently being designed, the workflow will basically be the same as before (section 2). However, major changes to the pilot classes are expected in order to facilitate the introduction of new features as well as to improve the overall organization of the code. The approach is to refactor the code gradually and introduce several new classes.

Currently the pilot is rather ATLAS specific. If PanDA and its pilot are used for other experiments it will be appropriate to have the code organized accordingly; to this end, a major new class will be the Experiment class, to which the ATLAS experiment will be a sub class [Fig. 2]. It will e.g. have a setup() method that defines how an ATLAS job should be setup, as well as other methods that are relevant to ATLAS. Any other PanDA system user can create a sub class of the Experiment class that will meet their demands of how a job should be setup, etc.

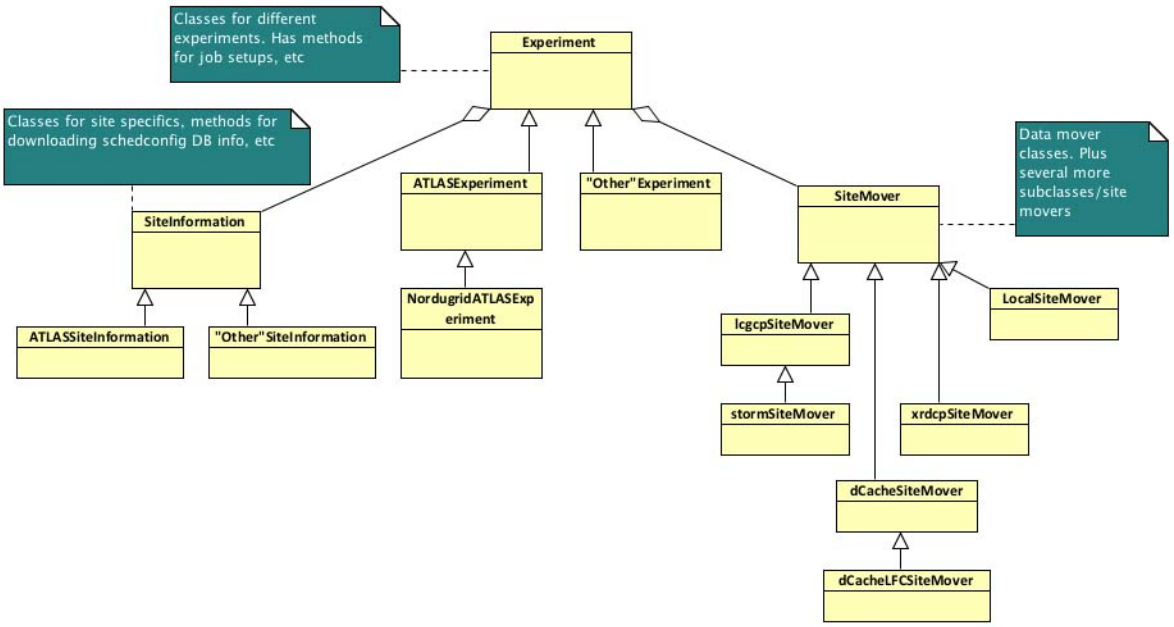

Figure 2. Diagram of some next generation PanDA pilot classes; SiteInformation, Experiment and SiteMover.

4. Recent improvements

The PanDA pilot is frequently updated with improvements and requests from the grid sites and software developers. The following sub sections describe a selection of recent improvements.

4.1. Job and transfer retries

Currently, the pilot can identify 100 different error types, The most common errors are grid related; failures during stage-in and stage-out due to temporary problems with the SE's. Other common errors include problems with software, local disk space, file system time-outs and issues with the file catalog. To reduce the impact of several of these problems, the pilot is equipped with multiple retry mechanisms. Stage-in and stage-out failures are automatically retried after a short delay. In the case of stage-in, the pilot can also try alternative replicas including replicas from a different SE.

In some 20+ failure cases among user analysis jobs, the pilot can instruct the PanDA server to automatically retry the entire job. Currently, the efficiency is 50% success after the first retry and 27% after the second retry (Fig. 3). The user job retry improvement is on the percent level. A later version of the retry mechanism is expected to include job retries at a different site.

Stage-out to a different SE (T1) and using Federated XrootD for more advanced alternative stage-in [10] are currently under development.

The pilot is also equipped with a job recovery mechanism. If a job fails to stage-out its output, the pilot can optionally leave the output files and the job log file on a local disk. When a later pilot arrives to the worker node it will find the remains of the previous job and will reattempt to stage-out the files.

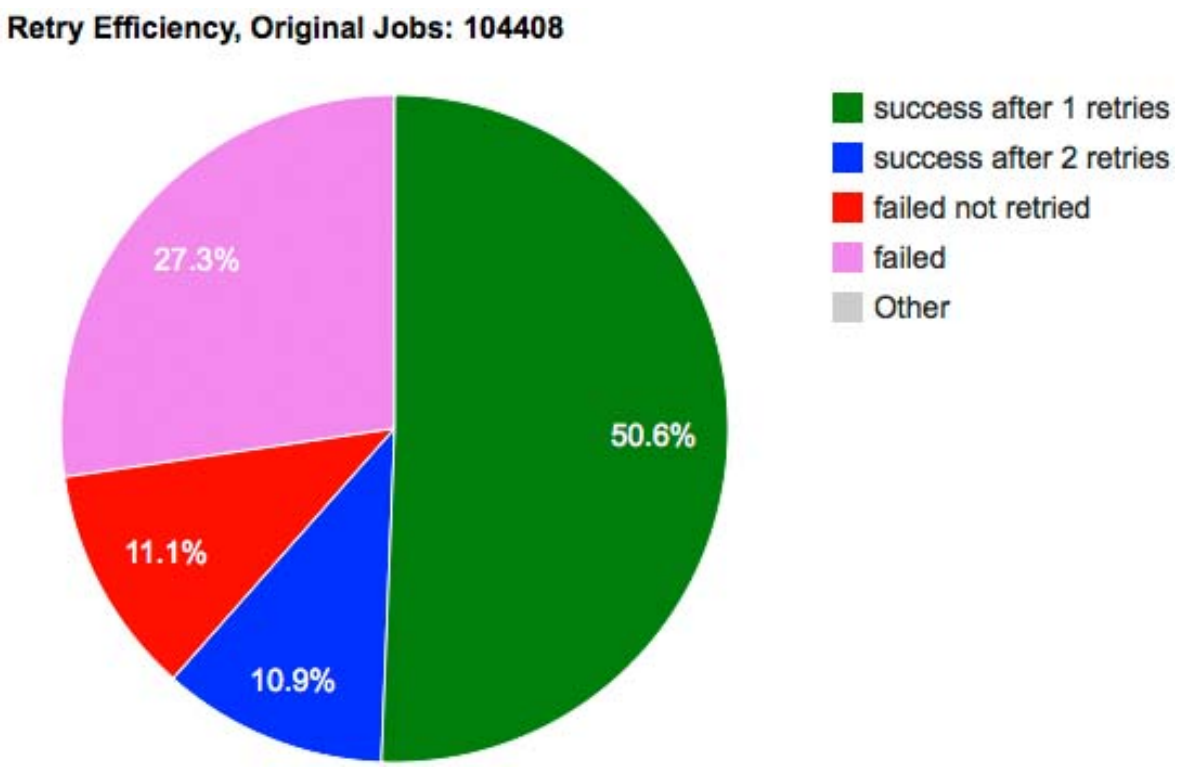

Figure 3. Efficiency of retried jobs.

4.2. CernVM File System integration

CernVM [11] is a baseline Virtual Software Appliance for the participants of the CERN LHC experiments, with a goal to remove a need for the installation of the experiment software and to minimize the number of platforms. Many ATLAS sites are already using the CernVM File System (CVMFS) to distribute releases, special files and other software (e.g. the PanDA pilot).

Since special files can be distributed with CVMFS, there is no need for the pilot to stage them in from the local SE. In the case that a job uses such a file, the pilot will recognize it and first look for it in a default CVMFS directory. If the file is available, the pilot will use it instead of staging it in.

4.3. JEM technology

The Job Execution Monitor [12] is a monitoring system running in user space. Users can submit their jobs with a special option that allows them to follow the job while it runs. JEM measures parameters like CPU load, incoming and outgoing network traffic, free RAM, free disk space on several file systems, etc [Fig. 4]. JEM can be reached from the standard PanDA monitor job page.

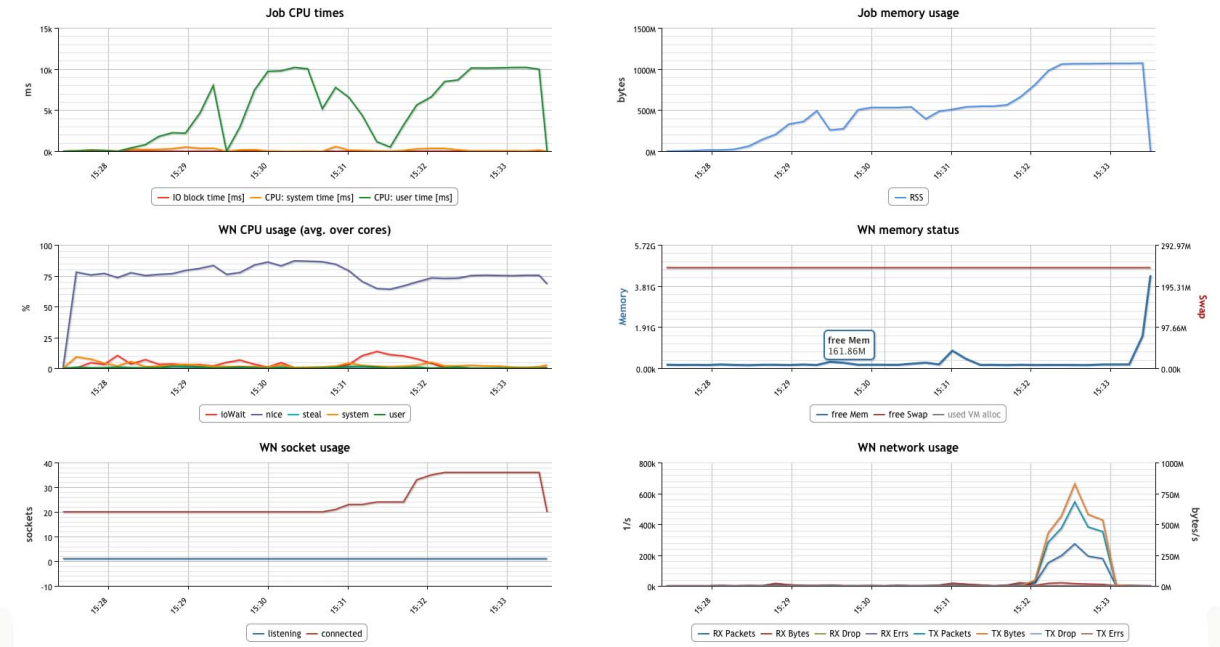

Figure 4. Example of monitoring diagrams from a PanDA jobs using JEM.

4.4. HammerCloud stress-testing system

HammerCloud [13] is a Distributed Analysis testing system used by several experiments. In ATLAS it is used not only to send standard test jobs to sites but also for sending special jobs for testing release candidates of the pilot.

The pilot wrappers randomly select the release candidate pilot version (on average once per one hundred pilot wrappers). The pilots ask the PanDA server for the specially labeled jobs created by HammerCloud. The progress can be followed on the HammerCloud and PanDA monitor pages. Most problems with the pilot are thus spotted before the development version of the pilot is released.

5. Performance

The PanDA system is concurrently serving up to 100k production jobs [Fig. 5] and 35k user analysis jobs [Fig. 6]. It is performing very well with high job efficiency. The error rate in the entire system is typically less than 10 percent. For production jobs the majority of the errors are site or system related, while the rest are problems with ATLAS software.

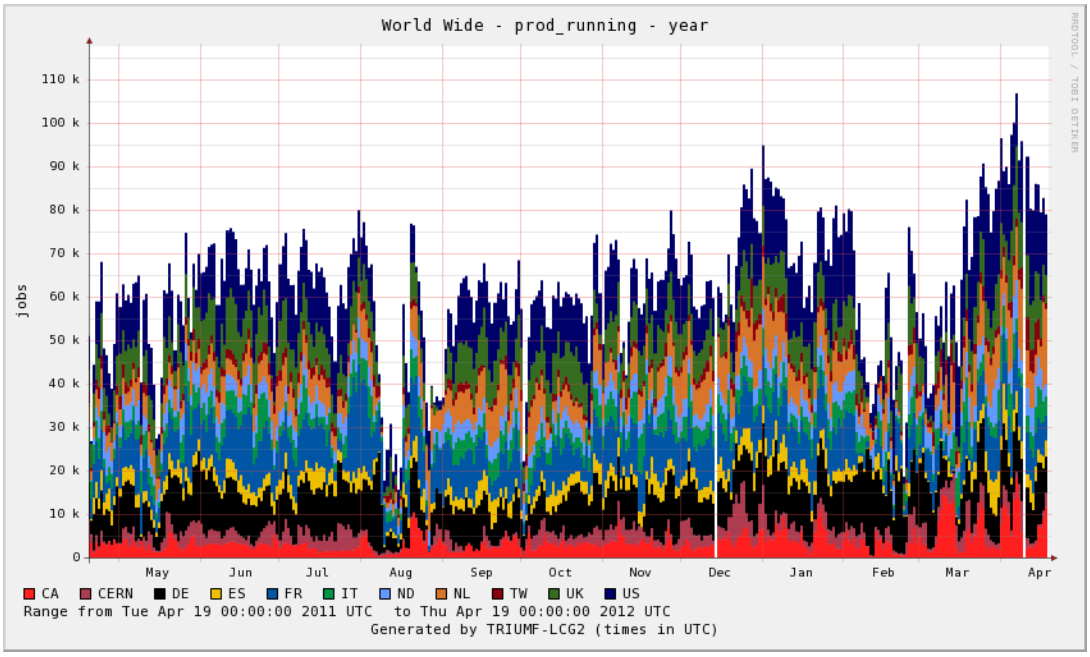

Figure 5. Concurrent production jobs in PanDA 2011-2012. Each coloured area represents the number of jobs in a certain cloud.

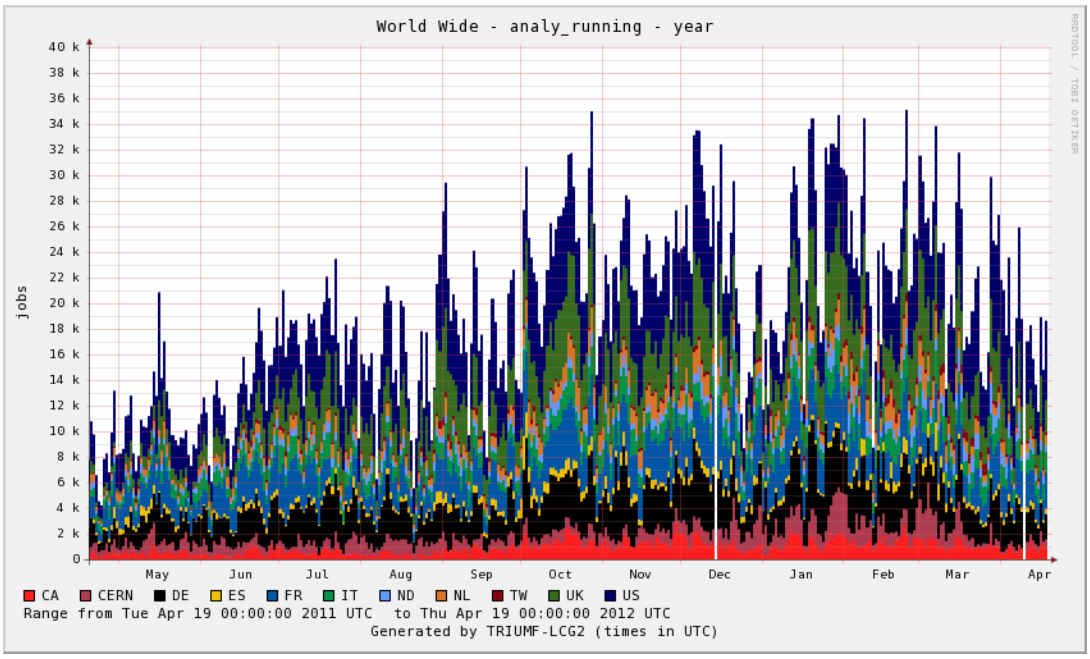

Figure 6. Concurrent user analysis jobs in PanDA 2011-2012. Each coloured area represents the number of jobs in a certain cloud.

6. Conclusions

The ATLAS Experiment has used the pilot-based PanDA system since 2005. It has proven to be very successful in managing the distributed production and analysis requirements on all three ATLAS grids, and is now being used to incorporate cloud computing resources into ATLAS computing. The

PanDA pilot is equipped with several advanced features, such as resilient stage-in and stage-out, entire job retries, JEM technology, as well as taking advantages of the benefits that the CernVM file system provides. The HammerCloud stress-testing system is used for convenient and automatic testing of new pilot versions. A new generation of the pilot is currently being designed which will alleviate the adoption of the PanDA system by other interested parties.

7. References

[1] The ATLAS Experiment: *ATLAS Technical Proposal*. ATLAS Collaboration. CERN/LHCC/94-43, 1994.

[2] J. Caballero, *AutoPyFactory: A Scalable Flexible Pilot Factory Implementation*, in Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP 2012).

[3] P. Nilsson, *The ATLAS PanDA Pilot in Operation*, Proc. of the 18th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2010)

[4] T. Maeno, *Overview of ATLAS PanDA Workload Management*, Proc. of the 18th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP 2010)

[5] Open Science Grid: http://www.opensciencegrid.org

[6] European Grid Initiative: http://www.egi.eu

[7] M. Ellert et al., *The Nordugrid Project*, NIM A, 2003, Vol. 502.

[8] F. Barreiro et al, *Exploiting Virtualization and Cloud Computing in ATLAS*, Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP 2012)

[9] PanDA Monitor: http://panda.cern.ch

[10] R. Gardner et al., *Using Xrootd to Federate Regional Storage*, Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP 2012)

[11] Predrag B., *Status and Future Perspectives of CernVM-FS*, in Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP 2012)

[12] JEM: http://jem.physik.uni-wuppertal.de/JEM

[13] HammerCloud: http://hammercloud.cern.ch